

N74-10176

**NASA TECHNICAL  
MEMORANDUM**



**NASA TM X-2901**

**NASA TM X-2901**

**CASE FILE  
COPY**

**INTERCOMPUTER TRANSFER IN FULL PRECISION  
OF ARBITRARY DATA ON MAGNETIC TAPE  
EMPLOYING NASTRAN USER TAPE FORMAT**

*by James L. Rogers, Jr.*

*Langley Research Center  
Hampton, Va. 23665*



1. Report No. NASA TM X-2901	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle INTERCOMPUTER TRANSFER IN FULL PRECISION OF ARBITRARY DATA ON MAGNETIC TAPE EMPLOYING NASTRAN USER TAPE FORMAT		5. Report Date November 1973	
		6. Performing Organization Code	
7. Author(s) James L. Rogers, Jr.		8. Performing Organization Report No. L-9052	
		10. Work Unit No. 501-22-01-01	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract  <p>A description is presented of two new utility programs which implement the transfer, in full precision, of arbitrary data (matrices or tables) between any of the three NASTRAN operative computers without the handling of large card decks. These computers include the CDC 6000 series, the IBM 360-370 series, and the UNIVAC 1100 series. The data may be generated by NASTRAN or by another computer program if the NASTRAN user tape format is employed.</p>			
17. Key Words (Suggested by Author(s)) Intercomputer data transfer NASTRAN		18. Distribution Statement Unclassified — Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 46	22. Price* Domestic, \$3.00 Foreign, \$5.50

# CONTENTS

	Page
SUMMARY . . . . .	1
INTRODUCTION . . . . .	1
OVERVIEW OF PROGRAMS . . . . .	2
DESCRIPTION OF RDUSER AND ITS SUBPROGRAMS . . . . .	4
Program RDUSER . . . . .	4
Subprogram RDCOM . . . . .	7
Subprogram MATRD . . . . .	8
Subprogram TABRD . . . . .	17
Subprogram INTGER . . . . .	21
RDUSER USAGE . . . . .	23
Control-Card Operation for RDUSER . . . . .	23
Error Messages Output by RDUSER . . . . .	24
Restrictions in RDUSER . . . . .	24
Card Input for RDUSER . . . . .	25
Sample Input for RDUSER . . . . .	25
DESCRIPTION OF WRTUSER AND ITS SUBPROGRAMS . . . . .	26
Program WRTUSER . . . . .	26
Subprogram CNVSRC . . . . .	29
Subprogram WRTCOM . . . . .	30
Subprogram HEADER . . . . .	31
Subprogram MATWRT . . . . .	32
Subprogram TABWRT . . . . .	36
WRTUSER USAGE . . . . .	38
Control-Card Operation for WRTUSER . . . . .	38
Error Messages Output by WRTUSER . . . . .	39
Restrictions in WRTUSER . . . . .	40
Card Input for WRTUSER . . . . .	40
Sample Input for WRTUSER . . . . .	40
VERIFICATION OF PROGRAMS . . . . .	40
CONCLUDING REMARKS . . . . .	41
REFERENCES . . . . .	41
TABLES . . . . .	42

INTERCOMPUTER TRANSFER IN FULL PRECISION OF  
ARBITRARY DATA ON MAGNETIC TAPE EMPLOYING  
NASTRAN USER TAPE FORMAT

By James L. Rogers, Jr.  
Langley Research Center

SUMMARY

A description is presented of two new utility programs which implement the transfer, in full precision, of arbitrary data (matrices or tables) between any of the three NASTRAN operative computers without the handling of large card decks. These computers include the CDC 6000 series, the IBM 360-370 series, and the UNIVAC 1100 series. The data may be generated by NASTRAN or by another computer program if the NASTRAN user tape format is employed.

INTRODUCTION

The NASTRAN computer program (refs. 1 and 2) is capable of execution on three different types of computers, namely, the CDC 6000 series, the IBM 360-370 series, and the UNIVAC 1100 series. A typical activity requiring transfer of data between dissimilar computers is the analysis of a large structure such as the Space Shuttle by substructuring. Models of portions of the vehicle which have been analyzed by subcontractors on their computers must be integrated into a model of the complete structure by the prime contractor on his computer. Presently the transfer of NASTRAN matrices or tables between two different types of computers is accomplished by punched cards or a magnetic tape containing card images. These methods of data transfer do not satisfy the requirements for intercomputer data transfer associated with a substructuring activity because (1) accuracy will be lost due to the precision limitations (10 significant digits) of the NASTRAN Direct Matrix Input (DMI) punched card and (2) large order matrices make card handling too cumbersome.

To provide a more satisfactory transfer of data, two new programs, RDUSER and WRTUSER, were created. These two programs, used in conjunction with NASTRAN modules OUTPUT2 (ref. 1, p. 5.3-20h) and INPUTT2 (ref. 1, p. 5.3-16h) available in level 15 and later versions of NASTRAN, allow data to be transferred between computers without loss of accuracy and without handling large decks of punched cards. The purpose of this paper is to describe both the utility programs RDUSER and WRTUSER and their applica-



tions by typical data transfer. Although data may come from any computer program using the NASTRAN user tape format, examples in this paper are confined to NASTRAN data since RDUSER and WRTUSER were written with the NASTRAN user in mind.

## OVERVIEW OF PROGRAMS

Beginning with level 15 NASTRAN provided the capability of using FORTRAN WRITE statements to write intermediate data blocks (matrices or tables) on a magnetic tape. This was made possible by the NASTRAN module OUTPUT2 which has the following calling sequence:

OUTPUT2 DB1, DB2, DB3, DB4, DB5//V, N, P1/V, N, P2/V, N, P3 \$

where the DBi are the data blocks to be written on tape, P1 is a parameter for positioning the tape, P2 is the FORTRAN unit number assigned to the tape, and P3 is the FORTRAN User Tape Label (default = XXXXXXXXX).

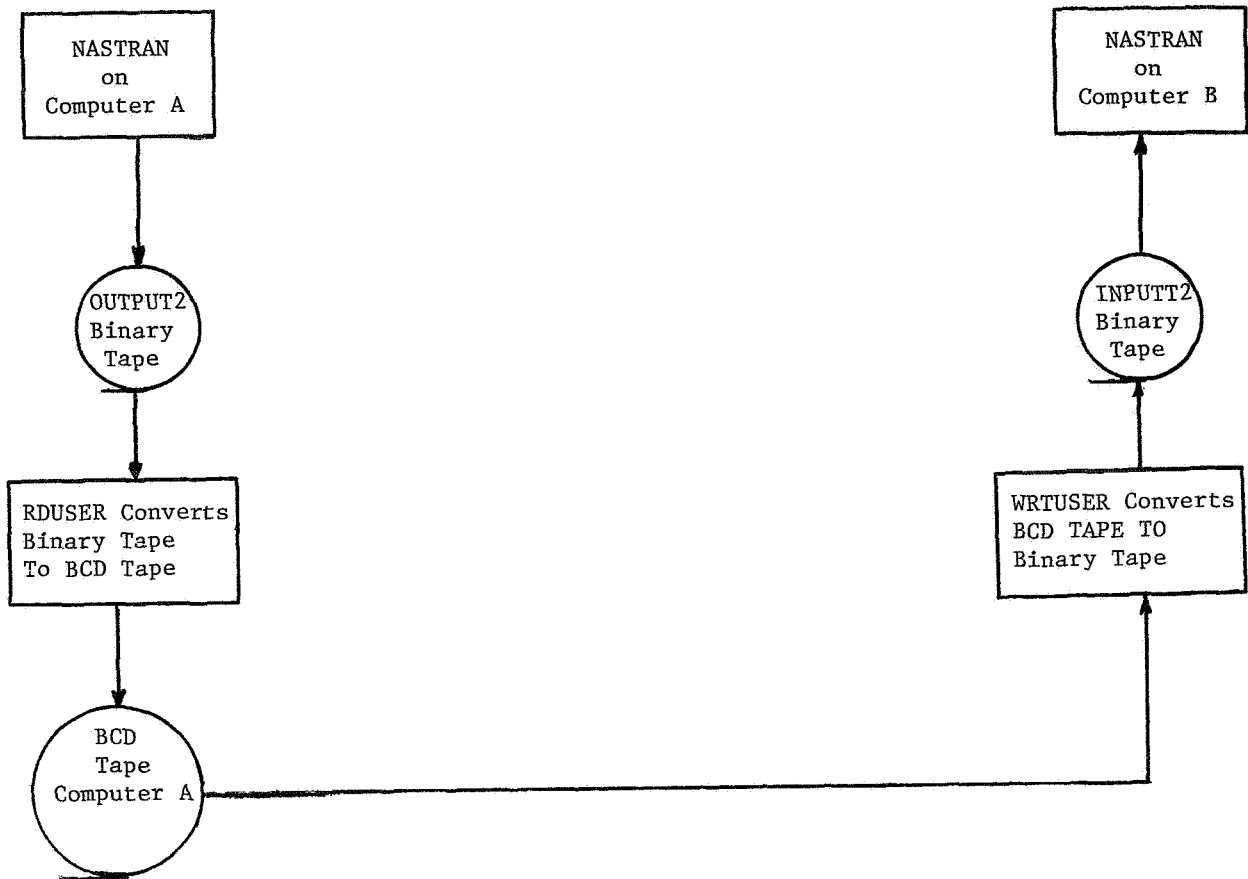
The tape created by OUTPUT2 is a binary tape. Its format is shown in tables I, II, and III. Tapes created by a program other than NASTRAN are acceptable as long as the data are output in this format. In order to write the header information on the tape, the P1 parameter must be -1 (rewind before writing) the first time OUTPUT2 is called in NASTRAN; otherwise, P1 is zero. This binary tape must be converted to a BCD (binary coded decimal) tape before it can be used on a computer of a different type. The conversion is performed by the utility program RDUSER which accepts tables and single- or double-precision, real or complex matrices. No precision is lost in generating the BCD tape, and the problem of handling large numbers of punched cards is alleviated.

The tape containing the BCD data is transferred to another installation. Before these data can be used as input for NASTRAN at this installation, two tasks must be performed. The first task is to convert the source of the BCD tape, written by RDUSER, to another source form readable by the computer on which the data will be used. The second task is to convert the BCD tape into an acceptable binary form for the NASTRAN module INPUTT2. The program WRTUSER accomplishes both of these tasks. The calling sequence for the INPUTT2 module has the form

INPUTT2 DB1, DB2, DB3, DB4, DB5/V, N, P1/V, N, P2/V, N, P3 \$

where the DBi are the data blocks to be recovered from the binary tape, P1 is a parameter for positioning the tape (P1 must be -1 for the first call to INPUTT2 and zero on all succeeding calls), P2 is the FORTRAN unit number assigned to the binary tape, and P3 is the FORTRAN User Tape Label (default = XXXXXXXXX). A flow chart of the complete tape interface method is as follows:



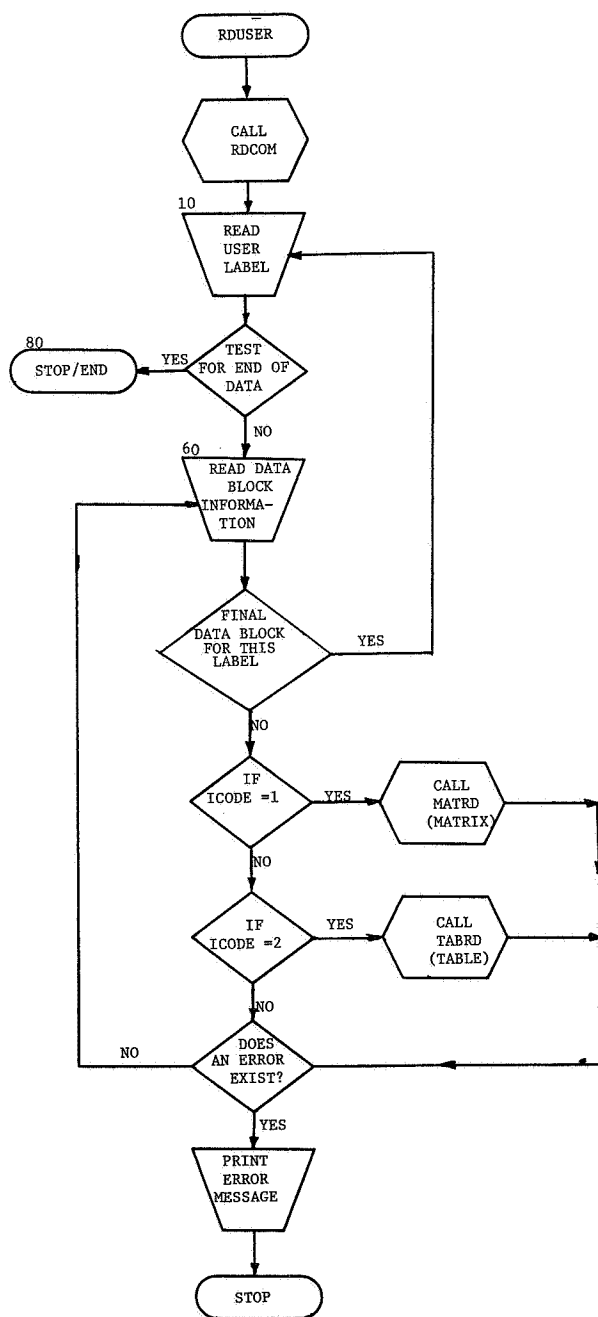




## DESCRIPTION OF RDUSER AND ITS SUBPROGRAMS

### Program RDUSER

RDUSER is a FORTRAN main program that has as its primary function the calling of various subprograms for data manipulation. The flow chart and the program listing that follow show how RDUSER controls program operation.





C		RDU	11
C	THIS PROGRAM CONVERTS A BINARY UNFORMATTED TAPE OUTPUT FROM	RDU	12
C	NASTRAN MODULE OUTPUT2 INTO A BCD TAPE	RDU	13
C		RDU	14
	DIMENSION IRDIN(7)	RDU	20
	DATA IBLANK,ISTOP,IT/4H ,4HSTOP,4H IT /	RDU	30
	WRITE (6,160)	RDU	40
C		RDU	50
C	COMENT READS USER INFORMATION ABOUT DATA BLOCKS ON TAPE	RDU	60
C		RDU	70
	CALL RDCOM	RDU	80
C		RDU	90
C	READ IN TAPE ID	RDU	100
C		RDU	110
10	READ (5,90) NAM1,NAM2	RDU	120
	IF (NAM1.EQ.IBLANK.AND.NAM2.EQ.IBLANK) GO TO 80	RDU	130
	ISW=0	RDU	131
C		RDU	140
C	READ TAPE UNTIL MATCHING ID IS FOUND	RDU	150
C		RDU	160
20	READ (9) ICNT	RDU	170
	IF (ISW.EQ.0.AND.ICNT.NE.3) GO TO 70	RDU	180
	ISW=1	RDU	181
	READ (9) (IRDIN(I),I=1,ICNT)	RDU	190
C		RDU	210
C	WRITE INFORMATION ON BCD TAPE	RDU	220
C		RDU	230
	IF (ICNT.EQ.3) WRITE (8,100) (IRDIN(I),I=1,ICNT)	RDU	240
C		RDU	250
C	WRITE INFORMATION ON BCD TAPE	RDU	260
C		RDU	270
	IF (ICNT.EQ.7) WRITE (8,110) (IRDIN(I),I=1,ICNT)	RDU	280
C		RDU	290
C	WRITE INFORMATION ON BCD TAPE	RDU	300
C		RDU	310
	IF (ICNT.EQ.2) WRITE (8,120) (IRDIN(I),I=1,ICNT)	RDU	320
C		RDU	330
C	CHECK FOR MATCH	RDU	340
C		RDU	350
	IF (NAM1.EQ.IRDIN(1).AND.NAM2.EQ.IRDIN(2)) GO TO 50	RDU	360
	GO TO 20	RDU	370
50	READ (9) IDUM	RDU	380
	READ (9) IDUM	RDU	390
C		RDU	400
C	ID HAS BEEN FOUND ON TAPE	RDU	410
C	ICODE=1 WE ARE LOOKING FOR A MATRIX	RDU	420
C	ICODE=2 WE ARE LOOKING FOR A TABLE	RDU	430
C	IOPT=0 DO NOT PRINT DATA BLOCK ELEMENTS	RDU	440
C	IOPT=1 PRINT DATA BLOCK ELEMENTS	RDU	450
C		RDU	460
60	READ (5,90) NAM1,NAM2,ICODE,IOPT	RDU	470
	IF (NAM1.EQ.IBLANK.AND.NAM2.EQ.IBLANK) GO TO 10	RDU	480
	IF (ICODE.EQ.1) CALL MATRD (NAM1,NAM2,IERR,IOPT)	RDU	490
	IF (ICODE.EQ.2) CALL TABRD (NAM1,NAM2,IERR,IOPT)	RDU	500
	IF (IERR.EQ.0) GO TO 60	RDU	510
C		RDU	520
C	MATRIX OR TABLE NAME NOT FOUND ON TAPE	RDU	530
C		RDU	540
	WRITE (6,130) NAM1,NAM2	RDU	550
	GO TO 80	RDU	560

C		RDU 570
C	TAPE ID WAS NOT FOUND ON TAPE	RDU 580
C		RDU 590
70	WRITE (6,140) NAM1,NAM2	RDU 600
	STOP	RDU 601
C		RDU 607
C	WRITE EOF ON TAPE	RDU 608
C		RDU 609
80	ICNT=0	RDU 610
	WRITE (8,150) ICNT	RDU 620
	WRITE (8,170) ISTOP,IT	RDU 621
	WRITE (6,160)	RDU 630
	STOP	RDU 640
C		RDU 650
90	FORMAT (2A4,2I2)	RDU 660
100	FORMAT (50X,3I10)	RDU 670
110	FORMAT (10X,7(A4,6X))	RDU 680
120	FORMAT (60X,2(A4,6X))	RDU 690
130	FORMAT (10H THE NAME ,2A4 ,23H WAS NOT FOUND ON TAPE /)	RDU 700
140	FORMAT (7H LABEL ,2A4 ,23H WAS NOT FOUND ON TAPE /)	RDU 710
150	FORMAT (70X,I10)	RDU 720
160	FORMAT (1H1)	RDU 730
170	FORMAT (72X,2A4)	RDU 731
	END	RDU 740-



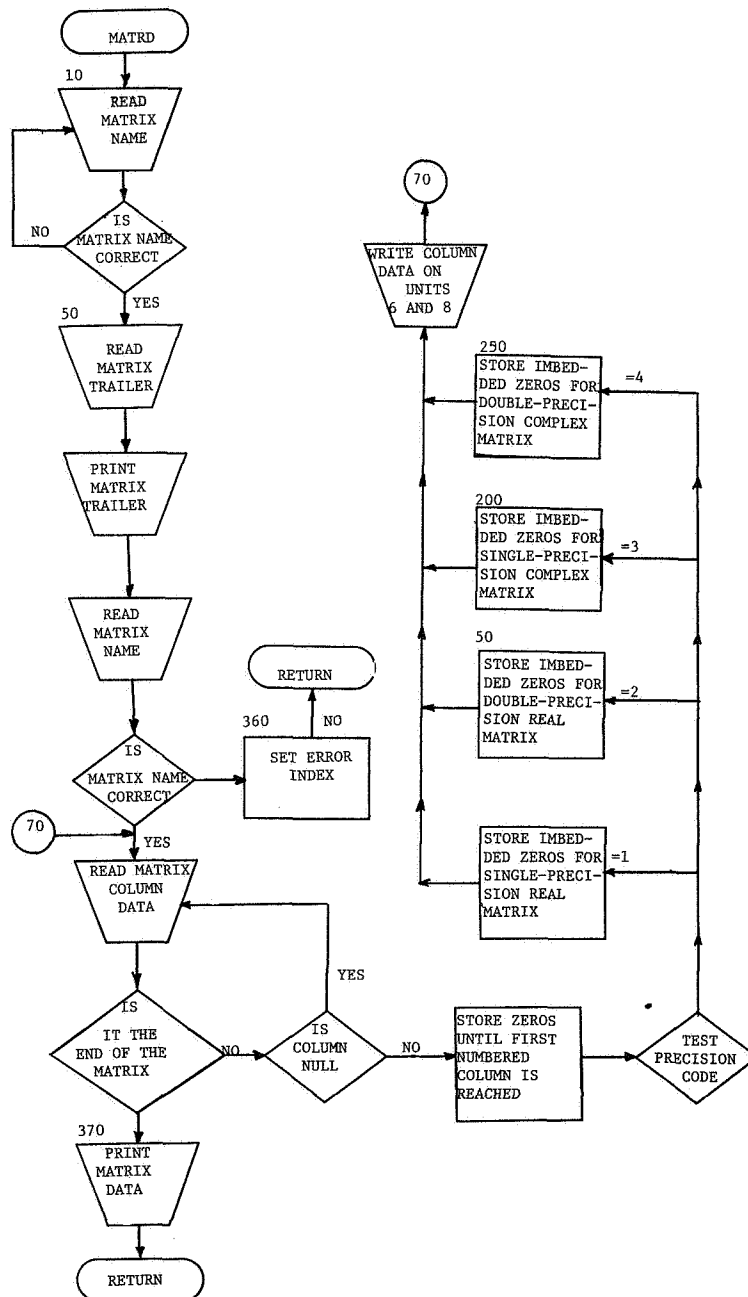
## Subprogram RDCOM

RDCOM is a FORTRAN subprogram. The primary job of RDCOM is to read comments by the user about the data on the tape. These comments are punched on cards for input into RDUSER. The cards are read with a free field format and thus may contain any information describing the data blocks. The comments are written on unit 8. The RDCOM subprogram listing follows.

	SUBROUTINE RDCOM	COM 10
C		COM 11
C	THIS SUBROUTINE READS COMMENTS FROM CARDS AND WRITES THEM ON TAPE	COM 12
C		COM 13
	DIMENSION ICOM(20)	COM 20
	DATA IQUIT/4H END/	COM 30
10	READ (5,20) (ICOM(I),I=1,20)	COM 40
	PRINT 20, (ICOM(I),I=1,20)	COM 50
	WRITE (8,20) (ICOM(I),I=1,20)	COM 60
	IF (ICOM(20).NE.IQUIT) GO TO 10	COM 70
	RETURN	COM 80
C		COM 90
20	FORMAT (20A4)	
	END	COM 110~

## Subprogram MATRD

MATRD is a FORTRAN subprogram. Its parameters are the matrix name (NAM1, NAM2), an error flag (IERR), and a print option (IOPT) for the user to print or not to print the elements of a column. The primary job of MATRD is to read the matrix data from unit 9, unpack the columns, and write them on unit 8 in a suitable format. If the matrix name cannot be found on unit 9, IERR will be set to one (1) and the program will stop upon returning to RDUSER. A MATRD flow chart and the subprogram listing follow.



	SUBROUTINE MATRD (NAM1,NAM2,IERR,IOPT)	MAT 10
C		MAT 11
C	THIS SUBROUTINE READS MATRIX DATA BLOCKS	MAT 12
C		MAT 13
C		MAT 14
C	DIMENSION STATEMENTS MUST BE CHANGED IN ACCORDANCE WITH	MAT 15
C	THE SIZE OF THE MATRICES BEING READ	MAT 16
C		MAT 17
	DOUBLE PRECISION DVAL(100)	MAT 20
	DIMENSION WRDIN(100), IRDIN(100), VAL(100)	MAT 30
	EQUIVALENCE (WRDIN(1),IRDIN(1))	MAT 40
	LOGICAL INTGER	MAT 50
	DATA ALPI,SQ1,SQ2,REC1,REC2,SYM1,SYM2,RE,COM1,COM2/1HI,4H SQU,4HARMAT 60	
	1E ,4HRECT,4HNGLE,4HSYME,4HTRIC,4HREAL,4HCOMP,4HLEX /	MAT 70
	DATA BLANK/4H /	MAT 80
	DATA SING1,SING2,DOUB1,DOUB2/4H SIN,4HGLE ,4H DOU,4HBLE /	MAT 90
C		MAT 100
C	*****	MAT 110
C	FORMAT FOR A MATRIX DATA BLOCK	MAT 120
C		MAT 130
C	RECORD NO. * WORD NO. * DESCRIPTION * TYPE	MAT 140
C	1 * 1-2 * MATRIX LABEL * ALPHA-NUMERIC * MAT 150	
C	2 * 1 * EOR (-1) * INTEGER * MAT 160	
C	3 * 1 *NO. WORDS NEXT REC. * INTEGER * MAT 170	
C	4 * 1 *(TRAILER) GINO NAME * INTEGER * MAT 180	
C	4 * 2 * NO. OF COLUMNS * INTEGER * MAT 190	
C	4 * 3 * NO. OF ROWS * INTEGER * MAT 200	
C	4 * 4 * FORM OF MATRIX * INTEGER * MAT 210	
C	4 * 5 * TYPE OF MATRIX * INTEGER * MAT 220	
C	4 * 6 *NO. NONZERO TERMS * INTEGER * MAT 230	
C	4 * 7 * PER CENT FULLNESS * INTEGER * MAT 240	
C	5 * 1 * EOR (-2) * INTEGER * MAT 250	
C	6 * 1 *NO. WORDS NEXT REC. * INTEGER * MAT 260	
C	7 * 1-2 * MATRIX LABEL * ALPHA-NUMERIC * MAT 270	
C	8 * 1 * EOR (-3) * INTEGER * MAT 280	
C	9 * 1 *NO. WORDS NEXT REC. * INTEGER * MAT 290	
C	10 * 1 *FIRST NON-ZERO COL. * INTEGER * MAT 300	
C	10 * 2 * PRECISION * INTEGER * MAT 310	
C	10 * 3 * NOT USED * INTEGER * MAT 320	
C	10 * 4 * NOT USED * INTEGER * MAT 330	
C	10 * 5 * NOT USED * INTEGER * MAT 340	
C	10 * 6-NO. WORDS-1* ELEMENTS OF ROW * REAL * MAT 350	
C	10 * 6-NO. WORDS-1*POINTERS ARE IMBEDDED* INTEGER * MAT 360	
C	10 * NO. WORDS * END OF COLUMN * MAT 370	
C		
C	RECORDS 8,9, AND 10 ARE REPEATED FOR EACH COLUMN OF THE MATRIX	MAT 380
C	WITH THE RECORD CORRESPONDING TO RECORD 8 DECREASING BY ONE	MAT 390
C	EACH TIME. THIS IS STOPPED WHEN A ZERO (0) IS ENCOUNTERED IN	MAT 400
C	THE RECORD CONTAINING THE NUMBER OF WORDS IN THE NEXT RECORD.	MAT 410
C		MAT 420
C	*****	MAT 430
C		MAT 440
	ICODE=1	MAT 450
	WRITE (6,380)	MAT 460
	IERR=0	MAT 470
C		MAT 480
C	READ TAPE	MAT 490
C		MAT 500
10	READ (9) ICNT	MAT 510
	IF (ICNT.EQ.0) GO TO 360	MAT 511
	READ (9) (IRDIN(I),I=1,ICNT)	MAT 530



C		MAT 550
C	CHECK FOR CORRECT MATRIX LABEL	MAT 560
C		MAT 570
C	WRITE INFORMATION ON BCD TAPE	MAT 580
	IF (NAM1.EQ.IRDIN(1).AND.NAM2.EQ.IRDIN(2)) GO TO 50	MAT 590
40	READ (9) JCNT	MAT 600
	READ (9) ICNT	MAT 610
	IF (ICNT.EQ.0) GO TO 10	MAT 620
	READ (9) (IRDIN(I),I=1,ICNT)	MAT 630
	GO TO 40	MAT 640
50	IF (ICNT.EQ.2) WRITE (8,390) (IRDIN(I),I=1,ICNT),ICODE	MAT 650
	READ (9) JCNT	MAT 660
	READ (9) ICNT	MAT 670
C		MAT 680
C	READ INFORMATION FROM TRAILER	MAT 690
C		MAT 700
	READ (9) (IRDIN(I),I=1,ICNT)	MAT 710
C		MAT 720
C	WRITE INFORMATION ON BCD TAPE	MAT 730
C		MAT 740
	WRITE (8,400) (IRDIN(I),I=1,ICNT)	MAT 750
	ICOL=IRDIN(2)	MAT 760
	IROW=IRDIN(3)	MAT 770
	IFORM=IRDIN(4)	MAT 780
	ITYPE=IRDIN(5)	MAT 790
	NUMNZ=IRDIN(6)	MAT 800
	FULL=FLOAT(IRDIN(7))/100.	MAT 810
	ICROW=2*IROW	MAT 820
C		MAT 830
C	PRINT INFORMATION FROM TRAILER	MAT 840
C		MAT 850
	WRITE (6,410) NAM1,NAM2,(IRDIN(IJ),IJ=1,ICNT)	MAT 860
	IF (IFORM.EQ.1) FORM1=SQ1	MAT 870
	IF (IFORM.EQ.1) FORM2=SQ2	MAT 880
	IF (IFORM.EQ.2) FORM1=REC1	MAT 890
	IF (IFORM.EQ.2) FORM2=REC2	MAT 900
	IF (IFORM.EQ.6) FORM1=SYM1	MAT 910
	IF (IFORM.EQ.6) FORM2=SYM2	MAT 920
	IF (ITYPE.EQ.1.OR.ITYPE.EQ.2) TYPE1=RE	MAT 921
	IF (ITYPE.EQ.1.OR.ITYPE.EQ.2) TYPE2=BLANK	MAT 922
	IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) TYPE1=COM1	MAT 923
	IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) TYPE2=COM2	MAT 924
C		MAT 930
C	PRINT MATRIX HEADER	MAT 940
C		MAT 950
	WRITE (6,420) NAM1,NAM2,TYPE1,TYPE2,ICOL,IROW,FORM1,FORM2	MAT 960
C		MAT 970
C	READ TAPE	MAT 980
C		MAT 990
60	READ (9) JCNT	MAT1000
	READ (9) ICNT	MAT1010
	READ (9) (IRDIN(I),I=1,ICNT)	MAT1020
C		MAT1030
C	CHECK FOR CORRECT MATRIX LABEL	MAT1040
C		MAT1050
	IF (NAM1.EQ.IRDIN(1).AND.IRDIN(2).EQ.NAM2) GO TO 65	MAT1060
	GO TO 360	MAT1070
65	WRITE (8,460) ICNT	MAT1071
	WRITE (8,550) (IRDIN(I),I=1,ICNT)	MAT1072
C		MAT1080
C	READ TAPE	MAT1090
C		MAT1100

70	READ (9) JCNT	MAT1112
	NULCOL=-JCNT-2	MAT1120
	READ (9) ICNT	MAT1130
C		MAT1140
C	TEST FOR ZERO TO END THE READING OF THE MATRIX	MAT1150
C		MAT1160
	IF (ICNT.EQ.0) GO TO 370	MAT1170
C		MAT1180
C	TEST FOR NULL COLUMN	MAT1190
C		MAT1200
	IF (ICNT.NE.1) GO TO 80	MAT1210
	READ (9) INUMB	MAT1220
	WRITE (8,460) ICNT	MAT1230
	WRITE (8,460) INUMB	MAT1240
	IF (IOPT.EQ.1) PRINT 430, NULCOL	MAT1250
	GO TO 70	MAT1260
C		MAT1270
C	READ RECORDS OF INFORMATION	MAT1280
C		MAT1290
80	READ (9) (IRDIN(1),I=1,5),(WRDIN(1),I=6,ICNT)	MAT1300
	NZROW=IRDIN(1)	MAT1320
	IPREC=IRDIN(2)	MAT1330
	K=0	MAT1340
	IK=-3	MAT1350
C		MAT1360
C	STORE ZEROS IN APPROPRIATE LOCATIONS	MAT1370
C		MAT1380
	IF (NZROW.EQ.1) GO TO 100	MAT1390
C		MAT1400
C	STORE ZEROS IN THE BEGINNING OF THE COLUMN	MAT1410
C		MAT1420
	NZROW=NZROW-1	MAT1430
	K=-1	MAT1440
	DO 90 I=1,NZROW	MAT1450
	IK=IK+4	MAT1460
	K=K+2	MAT1470
	IF (IPREC.EQ.1) VAL(I)=0.	MAT1480
	IF (IPREC.EQ.2) DVAL(I)=0.000	MAT1490
	IF (IPREC.EQ.3) VAL(K)=0.	MAT1500
	IF (IPREC.EQ.3) VAL(K+1)=0.	MAT1510
	IF (IPREC.EQ.4) DVAL(K)=0.000	MAT1520
	IF (IPREC.EQ.4) DVAL(K+1)=0.000	MAT1530
C		MAT1540
C	STORE DATA FOR BCD TAPE	MAT1550
C		MAT1560
	IF (IPREC.EQ.2) VAL(K)=0.	MAT1570
	IF (IPREC.EQ.2) VAL(K+1)=0.	MAT1580
	IF (IPREC.EQ.4) VAL(IK)=0.	MAT1590
	IF (IPREC.EQ.4) VAL(IK+1)=0.	MAT1600
	IF (IPREC.EQ.4) VAL(IK+2)=0.	MAT1610
	IF (IPREC.EQ.4) VAL(IK+3)=0.	MAT1620
90	CONTINUE	MAT1630
	IF (IPREC.EQ.2) K=K+1	MAT1640
	NZROW=NZROW+1	MAT1650
100	I=NZROW-1	MAT1660
	IF (IPREC.EQ.3.OR.IPREC.EQ.4) I=2*(NZROW-1)-1	MAT1670
	IF (IPREC.EQ.2) GO TO 150	MAT1680
	IF (IPREC.EQ.3) GO TO 200	MAT1690
	IF (IPREC.EQ.4) GO TO 250	MAT1700
C		MAT1710
C	STORE SINGLE PRECISION MATRIX ELEMENTS	MAT1720
C		MAT1730

	IWRD=5	MAT1740
	J=NZROW+ICNT-7	MAT1750
110	I=I+1	MAT1760
	IWRD=IWRD+1	MAT1770
C		MAT1780
C	CHECK POINTER FOR IMBEDDED ZEROS	MAT1790
C		MAT1800
	IF (INTGER(WRDIN(IWRD))) GO TO 120	MAT1810
	GO TO 140	MAT1811
120	JJ=IRDIN(IWRD)-1	MAT1820
	I=I-1	MAT1830
130	I=I+1	MAT1840
C		MAT1850
C	STORE IMBEDDED ZEROS	MAT1860
C		MAT1870
	VAL(I)=0.	MAT1880
	IF (I.NE.JJ) GO TO 130	MAT1890
	I=I+1	MAT1900
	IWRD=IWRD+1	MAT1910
140	VAL(I)=WRDIN(IWRD)	MAT1920
	KCNT=IWRD+NZROW-6	MAT1930
	IF (KCNT.LT.J) GO TO 110	MAT1940
	GO TO 300	MAT1950
C		MAT1960
C	STORE DOUBLE PRECISION MATRIX ELEMENTS	MAT1970
C		MAT1980
150	IWRD=4	MAT1990
	IK=K-1	MAT2000
	J=NZROW+ICNT-6	MAT2010
160	I=I+1	MAT2020
	IWRD=IWRD+2	MAT2030
C		MAT2040
C	CHECK IMBEDDED ZEROS	MAT2050
C		MAT2060
	IF (INTGER(WRDIN(IWRD))) GO TO 170	MAT2070
	GO TO 190	MAT2071
170	JJ=IRDIN(IWRD)-1	MAT2080
	I=I-1	MAT2090
180	I=I+1	MAT2100
C		MAT2110
C	STORE IMBEDDED ZEROS	MAT2120
C		MAT2130
	DVAL(I)=0.0D0	MAT2140
C		MAT2150
C	STORE DATA FOR BCD TAPE	MAT2160
C		MAT2170
	IK=IK+2	MAT2180
	VAL(IK)=0.	MAT2190
	VAL(IK+1)=0.	MAT2200
	IF (I.NE.JJ) GO TO 180	MAT2210
	I=I+1	MAT2220
	IWRD=IWRD+1	MAT2230
190	DVAL(I)=DBLE(WRDIN(IWRD))	MAT2240
C		MAT2250
C	STORE DATA FOR BCD TAPE	MAT2260
C		MAT2270
	IK=IK+2	MAT2280
	VAL(IK)=WRDIN(IWRD)	MAT2290
	VAL(IK+1)=WRDIN(IWRD+1)	MAT2300
	KCNT=IWRD+NZROW-4	MAT2310
	IF (KCNT.LT.J) GO TO 160	MAT2320
	GO TO 300	MAT2330



C		MAT2340
C	STORE COMPLEX MATRIX ELEMENTS	MAT2350
C		MAT2360
200	IWRD=4	MAT2370
	J=NZROW+ICNT-6	MAT2380
210	I=I+2	MAT2390
	IWRD=IWRD+2	MAT2400
C		MAT2410
C	CHECK IMBEDDED ZEROS	MAT2420
C		MAT2430
	IF (INTGER(WRDIN(IWRD))) GO TO 220	MAT2440
	GO TO 240	MAT2441
220	IF (IRDIN(IWRD).EQ.0) GO TO 240	MAT2450
	JJ=2*(IRDIN(IWRD)-1)-1	MAT2460
	I=I-2	MAT2470
C		MAT2480
C	STORE IMBEDDED ZEROS	MAT2490
C		MAT2500
230	I=I+2	MAT2510
	VAL(I)=0.	MAT2520
	VAL(I+1)=0.	MAT2530
	IF (I.NE.JJ) GO TO 230	MAT2540
	I=I+2	MAT2550
	IWRD=IWRD+1	MAT2560
240	VAL(I)=WRDIN(IWRD)	MAT2570
	VAL(I+1)=WRDIN(IWRD+1)	MAT2580
	KCNT=IWRD+NZROW-4	MAT2590
	IF (KCNT.LT.J) GO TO 210	MAT2600
	GO TO 300	MAT2610
C		MAT2620
C	STORE COMPLEX DOUBLE PRECISION MATRIX ELEMENTS	MAT2630
C		MAT2640
250	IWRD=2	MAT2650
	J=NZROW+ICNT-4	MAT2660
260	I=I+2	MAT2670
	IWRD=IWRD+4	MAT2680
C		MAT2690
C	CHECK POINTER FOR IMBEDDED ZEROS	MAT2700
C		MAT2710
	IF (INTGER(WRDIN(IWRD))) GO TO 270	MAT2720
	GO TO 290	MAT2721
270	IF (IRDIN(IWRD).EQ.0) GO TO 290	MAT2730
	JJ=2*(IRDIN(IWRD)-1)-1	MAT2740
	I=I-2	MAT2750
C		MAT2760
C	STORE IMBEDDED ZEROS	MAT2770
C		MAT2780
280	I=I+2	MAT2790
	DVAL(I)=0.0D0	MAT2800
	DVAL(I+1)=0.0D0	MAT2810
C		MAT2820
C	STORE DATA FOR BCD TAPE	MAT2830
C		MAT2840
	IK=IK+4	MAT2850
	VAL(IK)=0.	MAT2860
	VAL(IK+1)=0.	MAT2870
	VAL(IK+2)=0.	MAT2880
	VAL(IK+3)=0.	MAT2890
	IF (I.NE.JJ) GO TO 280	MAT2900
	I=I+2	MAT2910
	IWRD=IWRD+1	MAT2920
290	DVAL(I)=DBLE(WRDIN(IWRD))	MAT2930

	DVAL(I+1)=DBLE(WRDIN(IWRD+2))	MAT2940
C		MAT2950
C	STORE DATA FOR BCD TAPE	MAT2960
C		MAT2970
	IK=IK+4	MAT2980
	VAL(IK)=WRDIN(IWRD)	MAT2990
	VAL(IK+1)=WRDIN(IWRD+1)	MAT3000
	VAL(IK+2)=WRDIN(IWRD+2)	MAT3010
	VAL(IK+3)=WRDIN(IWRD+3)	MAT3020
	KCNT=IWRD+NZROW	MAT3030
	IF (KCNT.LT.J) GO TO 260	MAT3040
300	IF ((IPREC.EQ.1.OR.IPREC.EQ.2).AND.IROW.EQ.I) GO TO 320	MAT3050
	IF (IPREC.EQ.1.OR.IPREC.EQ.2) K=(I+1)/2	MAT3060
	IF ((IPREC.EQ.3.OR.IPREC.EQ.4).AND.IROW.EQ.K) GO TO 320	MAT3070
	IF (IPREC.EQ.3.OR.IPREC.EQ.4) K=I	MAT3080
C		MAT3090
C	STORE ZEROS AT THE END OF THE COLUMN	MAT3100
C		MAT3110
	J=I+1	MAT3120
	DO 310 I=J,IROW	MAT3130
	IF (IPREC.EQ.2) IK=IK+2	MAT3140
	IF (IPREC.EQ.4) IK=IK+4	MAT3150
	K=K+2	MAT3160
	IF (IPREC.EQ.1) VAL(I)=0.	MAT3170
	IF (IPREC.EQ.2) DVAL(I)=0.0D0	MAT3180
	IF (IPREC.EQ.3) VAL(K)=0.	MAT3190
	IF (IPREC.EQ.3) VAL(K+1)=0.	MAT3200
	IF (IPREC.EQ.4) DVAL(K)=0.0D0	MAT3210
	IF (IPREC.EQ.4) DVAL(K+1)=0.0D0	MAT3220
C		MAT3230
C	STORE DATA FOR BCD TAPE	MAT3240
C		MAT3250
	IF (IPREC.EQ.2) VAL(IK)=0.	MAT3260
	IF (IPREC.EQ.2) VAL(IK+1)=0.	MAT3270
	IF (IPREC.EQ.4) VAL(IK)=0.	MAT3280
	IF (IPREC.EQ.4) VAL(IK+1)=0.	MAT3290
	IF (IPREC.EQ.4) VAL(IK+2)=0.	MAT3300
	IF (IPREC.EQ.4) VAL(IK+3)=0.	MAT3310
310	CONTINUE	MAT3320
320	IF (IOPT.EQ.1) PRINT 440, NULCOL	MAT3330
	IF (IPREC.EQ.2) GO TO 330	MAT3340
	IF (IPREC.EQ.3) GO TO 340	MAT3350
	IF (IPREC.EQ.4) GO TO 350	MAT3360
C		MAT3370
C	PRINT SINGLE PRECISION NUMBERS	MAT3380
C		MAT3390
	IF (IOPT.EQ.1) WRITE (6,450) (VAL(I),I=1,IROW)	MAT3400
C		MAT3410
C	WRITE INFORMATION ON BCD TAPE	MAT3420
C		MAT3430
	WRITE (8,460) IROW	MAT3440
C		MAT3450
C	WRITE INFORMATION ON BCD TAPE	MAT3460
C		MAT3470
	WRITE (8,470) (IRDIN(I),I=1,5)	MAT3480
C		MAT3490
C	WRITE INFORMATION ON BCD TAPE	MAT3500
C		MAT3510
	WRITE (8,480) (VAL(I),I=1,IROW)	MAT3520
	GO TO 70	MAT3530
C		MAT3540
C	PRINT DOUBLE PRECISION NUMBERS	MAT3550
C		MAT3560

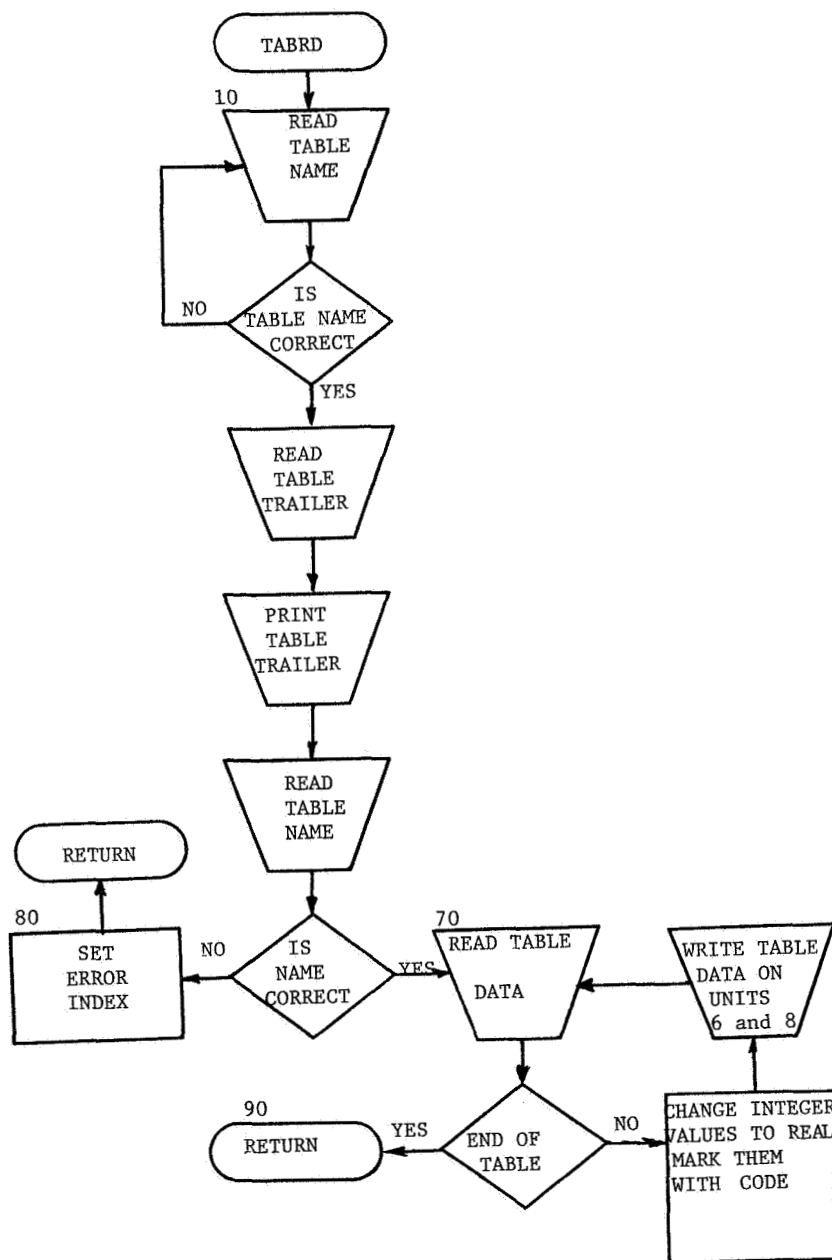
330	IF (IOPT.EQ.1) WRITE (6,490) (DVAL(I),I=1,IROW)	MAT3570
	IROW1=IROW*2	MAT3580
C		MAT3590
C	WRITE INFORMATION ON BCD TAPE	MAT3600
C		MAT3610
	WRITE (8,460) IROW1	MAT3620
C		MAT3630
C	WRITE INFORMATION ON BCD TAPE	MAT3640
C		MAT3650
	WRITE (8,470) (IRDIN(I),I=1,5)	MAT3660
C		MAT3670
C	WRITE INFORMATION ON BCD TAPE	MAT3680
C		MAT3690
	WRITE (8,480) (VAL(I),I=1,IROW1)	MAT3700
	GO TO 70	MAT3710
C		MAT3720
C	PRINT COMPLEX SINGLE PRECISION NUMBERS	MAT3730
C		MAT3740
340	IF (IOPT.EQ.1) WRITE (6,500) (VAL(I),VAL(I+1),ALPI,I=1,ICROW,2)	MAT3750
C		MAT3760
C	WRITE INFORMATION ON BCD TAPE	MAT3770
C		MAT3780
	WRITE (8,460) ICROW	MAT3790
C		MAT3800
C	WRITE INFORMATION ON BCD TAPE	MAT3810
C		MAT3820
	WRITE (8,470) (IRDIN(I),I=1,5)	MAT3830
C		MAT3840
C	WRITE INFORMATION ON BCD TAPE	MAT3850
C		MAT3860
	WRITE (8,480) (VAL(I),I=1,ICROW)	MAT3870
	GO TO 70	MAT3880
C		MAT3890
C	PRINT COMPLEX DOUBLE PRECISION NUMBERS	MAT3900
C		MAT3910
350	IF (IOPT.EQ.1) WRITE (6,510) (DVAL(I),DVAL(I+1),ALPI,I=1,ICROW,2)	MAT3920
	ICROW1=ICROW*2	MAT3930
C		MAT3940
C	WRITE INFORMATION ON BCD TAPE	MAT3950
C		MAT3960
	WRITE (8,460) ICROW1	MAT3970
C		MAT3980
C	WRITE INFORMATION ON BCD TAPE	MAT3990
C		MAT4000
	WRITE (8,470) (IRDIN(I),I=1,5)	MAT4010
C		MAT4020
C	WRITE INFORMATION ON BCD TAPE	MAT4030
C		MAT4040
	WRITE (8,480) (VAL(I),I=1,ICROW1)	MAT4050
	GO TO 70	MAT4060
C		MAT4061
C	RETURN WITH ERROR MESSAGE	MAT4062
C		MAT4063
360	IERR=1	MAT4070
	RETURN	MAT4080
C		MAT4090
C	PRINT MATRIX INFORMATION	MAT4100
C		MAT4110



370	WRITE (6,520) NUMNZ	MAT4120
	WRITE (6,530) FULL	MAT4130
	IF (IPREC.EQ.1.OR.IPREC.EQ.3) WRITE (6,540) SING1,SING2	MAT4140
	IF (IPREC.EQ.2.OR.IPREC.EQ.4) WRITE (6,540) DOUB1,DOUB2	MAT4150
C		MAT4160
C	WRITE INFORMATION ON BCD TAPE	MAT4170
	WRITE (8,460) ICNT	MAT4180
	RETURN	MAT4190
C		MAT4200
380	FORMAT (1H1)	MAT4210
390	FORMAT (50X,2(A4,6X),110)	MAT4220
400	FORMAT (10X,7I10)	MAT4230
410	FORMAT (//28H THE TRAILER FOR THE MATRIX ,2A4,4H IS ,7I7)	MAT4240
420	FORMAT (//8H MATRIX ,2A4,6H IS A ,2A4,14,10H COLUMN X ,14,5H ROW ,	MAT4250
	12A4,8H MATRIX /)	MAT4260
430	FORMAT (/8H COLUMN ,16,8H IS NULL/)	MAT4270
440	FORMAT (/8H COLUMN ,16)	MAT4280
450	FORMAT (6E20,12,1X/)	MAT4290
460	FORMAT (70X,110)	MAT4300
470	FORMAT (30X,5I10)	MAT4310
480	FORMAT (4E20,12)	MAT4320
490	FORMAT (4D30,22,1X/)	MAT4330
500	FORMAT (3(E20,12,E20,12,A1),1X/)	MAT4340
510	FORMAT (2(D30,22,D30,22,A1),1X/)	MAT4350
520	FORMAT (/54H THE NUMBER OF NON-ZERO WORDS IN THE LONGEST RECORD	MAT4360
	1,16)	MAT4370
530	FORMAT (/31H THE DENSITY OF THIS MATRIX IS ,F8,2,9H PERCENT /)	MAT4380
540	FORMAT (/15H THIS MATRIX IS,2A4,9HPRECISION)	MAT4390
550	FORMAT (2(A4,6X),(6I10))	MAT4391
	END	MAT4400-

### Subprogram TABRD

TABRD is a FORTRAN subprogram. Its parameters are the table name (NAM1, NAM2), an error flag (IERR), and a print option (IOPT) for the user to print or not to print the elements of the table. The primary tasks of TABRD are to read the table data from unit 9 and write it on unit 8 in a suitable format. TABRD converts integers into real numbers to facilitate the transfer. Integers are marked with a code so that they are easily recognized by the WRTUSER program. A TABRD flow chart and the subprogram listing follow.



C	SUBROUTINE TABRD (NAM1,NAM2,IERR,IOPT)	TAB 10
C		TAB 11
C	THIS SUBROUTINE READS TABLE DATA BLOCKS	TAB 12
C		TAB 13
C		TAB 14
C	DIMENSION STATEMENTS MUST BE CHANGED IN ACCORDANCE WITH	TAB 15
C	THE SIZE OF THE TABLES BEING READ	TAB 16
C		TAB 17
	DIMENSION IRDIN(100),WRDIN(100),VAL(100)	TAB 20
	EQUIVALENCE(IRDIN(1),WRDIN(1))	TAB 21
	LOGICAL INTGER	TAB 22
C		TAB 30
C	*****	TAB 40
C	FORMAT FOR A TABLE DATA BLOCK	TAB 50
C		TAB 60
C	RECORD NO. * WORD NO. * DESCRIPTION * TYPE	TAB 70
C	1 * 1-2 * TABLE LABEL * ALPHA-NUMERIC *	TAB 80
C	2 * 1 * EOR (-1) * INTEGER *	TAB 90
C	3 * 1 *NO. WORDS NEXT REC. * INTEGER *	TAB 100
C	4 * 1 * TRAILER GINO NAME * INTEGER *	TAB 110
C	4 * 2-N * MISC. INFORMATION * INTEGER *	TAB 120
C	5 * 1 * EOR (-2) * INTEGER *	TAB 130
C	6 * 1 *NO. WORDS NEXT REC. * INTEGER *	TAB 140
C	7 * 1-2 * TABLE LABEL * INTEGER *	TAB 150
C	8 * 1 * EOR (-3) * INTEGER *	TAB 160
C	9 * 1 *NO. WORDS NEXT REC. * INTEGER *	TAB 170
C	10 * ALL *ELEMENTS OF FIRST * INTEGER,REAL*	TAB 180
C	10 * ALL * RECORD OF TABLE * INTEGER,REAL*	TAB 190
C	RECORDS 8,9, AND 10 ARE REPEATED FOR EACH RECORD OF THE TABLE,	TAB 200
C	WITH THE RECORD CORRESPONDING TO RECORD 8 DECREASING BY ONE EACH	TAB 210
C	TIME. THIS IS STOPPED WHEN A ZERO (0) IS ENCOUNTERED IN THE RECORD	TAB 220
C	CONTAINING THE NUMBER OF WORDS IN THE NEXT RECORD.	TAB 230
C		TAB 240
C	*****	TAB 250
C		TAB 260
	ICODE=2	TAB 270
	WRITE (6,100)	TAB 280
	IERR=0	TAB 290
C		TAB 300
C	READ TAPE	TAB 310
C		TAB 320
10	READ (9) ICNT	TAB 330
	IF (ICNT.EQ.0) GO TO 80	TAB 331
	READ (9) (IRDIN(I),I=1,ICNT)	TAB 340
C		TAB 370
C	CHECK FOR CORRECT TABLE LABEL	TAB 380
C		TAB 390
	IF (NAM1.EQ.IRDIN(1).AND.NAM2.EQ.IRDIN(2)) GO TO 50	TAB 400
40	READ (9) JCNT	TAB 410
	READ (9) ICNT	TAB 420
	IF (ICNT.EQ.0) GO TO 10	TAB 430
	READ (9) (IRDIN(I),I=1,ICNT)	TAB 440
	GO TO 40	TAB 450
C		TAB 460
C	WRITE INFORMATION ON BCD TAPE	TAB 470
C		TAB 480
50	IF (ICNT.EQ.2) WRITE (8,110) (IRDIN(I),I=1,ICNT),ICODE	TAB 490
	READ (9) JCNT	TAB 500
	READ (9) ICNT	TAB 510
C		TAB 520
C	READ TRAILER	TAB 530
C		TAB 540



	READ (9) (IRDIN(I),I=1,ICNT)	TAB 550
C		TAB 560
C	WRITE INFORMATION ON BCD TAPE	TAB 570
C		TAB 580
	WRITE (8,120) (IRDIN(I),I=1,ICNT)	TAB 590
C		TAB 600
C	PRINT TRAILER	TAB 610
C		TAB 620
	WRITE (6,130) NAM1,NAM2,(IRDIN(I),I=1,ICNT)	TAB 630
C		TAB 640
C	READ TAPE	TAB 650
C		TAB 660
	READ (9) JCNT	TAB 670
	READ (9) ICNT	TAB 680
	READ (9) (IRDIN(I),I=1,ICNT)	TAB 690
C		TAB 700
C	CHECK FOR CORRECT TABLE LABEL	TAB 710
C		TAB 720
	IF (NAM1.EQ.IRDIN(1).AND.NAM2.EQ.IRDIN(2)) GO TO 60	TAB 730
	GO TO 80	TAB 740
60	NCNT=0	TAB 750
	WRITE (8,140) ICNT	TAB 751
	WRITE (8,180) (IRDIN(I),I=1,ICNT)	TAB 752
C		TAB 760
C	READ TAPE	TAB 770
C		TAB 780
70	READ (9) JCNT	TAB 790
	READ (9) ICNT	TAB 800
C		TAB 850
C	TEST FOR ZERO (0) TO END THE READING OF THE TABLE	TAB 860
C		TAB 870
	IF (ICNT.EQ.0) GO TO 90	TAB 880
	NCNT=NCNT+1	TAB 890
C		TAB 900
C	READ THE VALUES OF THE TABLE	TAB 910
C		TAB 920
	READ (9) (WRDIN(I),I=1,ICNT)	TAB 921
C		TAB 922
C	TEST ELEMENTS TO BE REAL OR INTEGER	TAB 923
C		TAB 924
	KCNT=0	TAB 925
	DO 75 I=1,ICNT	TAB 926
	IF (INTEGER(WRDIN(I)).AND.WRDIN(I).NE.0.0) GO TO 71	TAB 927
	GO TO 72	TAB 928
71	KCNT=KCNT+1	TAB 929
	VAL(KCNT)=-9999.99	TAB 930
C		TAB 931
C	TEMPORARILY CHANGE INTEGER TO REAL , STORE CODE TO SHOW CHANGE	TAB 932
C		TAB 933
	KCNT=KCNT+1	TAB 934
	VAL(KCNT)=FLOAT(IRDIN(I))	TAB 935
	GO TO 75	TAB 936
72	KCNT=KCNT+1	TAB 937
	VAL(KCNT)=WRDIN(I)	TAB 938
75	CONTINUE	TAB 939
C		TAB 940
C	WRITE INFORMATION ON BCD TAPE	TAB 950
C		TAB 960
	WRITE (8,140) KCNT	TAB 961
	WRITE (8,150) (VAL(I),I=1,KCNT)	TAB 970
C		TAB 980
C	PRINT THE VALUES OF THE TABLE	TAB 990
C		TAB1000

IF (IOPT.EQ.1) WRITE (6,160) NCNT,(IRDIN(I),I=1,ICNT)	TAB1010
IF (IOPT.EQ.1) WRITE (6,170) NCNT,(WRDIN(I),I=1,ICNT)	TAB1020
GO TO 70	TAB1030
80 IERR=1	TAB1040
C	TAB1041
C WRITE INFORMATION ON BCD TAPE	TAB1042
C	TAB1043
90 WRITE (8,140) ICNT	TAB1044
RETURN	TAB1050
C	TAB1060
100 FORMAT (1H1)	TAB1070
110 FORMAT (50X,2(A4,6X),I10)	TAB1080
120 FORMAT (10X,7I10)	TAB1090
130 FORMAT (//27H THE TRAILER FOR THE TABLE ,2A4,4H IS /(13I10))	TAB1100
140 FORMAT (70X,I10)	TAB1110
150 FORMAT (4E20,12)	TAB1120
160 FORMAT (/34H THE INTEGER VALUES OF RECORD NO. ,I6,4H ARE/, (13I10))	TAB1130
170 FORMAT (/31H THE REAL VALUES OF RECORD NO. ,I6,4H ARE/, (6E20,12))	TAB1131
180 FORMAT (2(A4,6X), (6I10))	TAB1132
END	TAB1140-

### Subprogram INTGER

INTGER is a logical function subprogram. It is written in FORTRAN or assembly language, depending upon the computer on which it is being used. It is a computer-dependent routine. INTGER has one parameter (WORD) which contains a number either real or integer. INTGER tests each of the exponent bits for a zero and returns a .TRUE. if each bit is a zero (indicating this number is an integer); otherwise it returns a .FALSE. (indicating this number is real). INTGER is called by both MATRD and TABRD. The INTGER subprogram listings for each of the three NASTRAN operative computers follow.

	IDENT	INTGER	INT	1
*			INT	2
*			INT	3
*	LOGICAL FUNCTION	INTGER (WORD)	INT	4
*			INT	5
*	RETURNS TRUE IF UPPER 12 BITS ARE ZERO		INT	6
*	FOR USE ON THE CDC 6000 SERIES COMPUTER		INT	7
*			INT	8
	ENTRY	INTGER	INT	9
	VFD	42/0LINTGER,18/2	INT	10
INTGER	BSS	1 E/E LINE	INT	11
	SA1	B1	INT	12
	MX2	12	INT	13
	BX3	X2*X1	INT	14
	MX6	60	INT	15
	ZR	X3,INTGER	INT	16
	SX6	B0	INT	17
	EQ	INTGER	INT	18
	END		INT	19

	LOGICAL FUNCTION	INTGER(WORD)	INT	10
C			INT	20
C	THIS SUBROUTINE TESTS THE UPPER EIGHT BITS OF A WORD FOR ZEROS		INT	30
C	IF ALL EIGHT BITS ARE ZERO INTGER RETURNS TRUE OTHERWISE FALSE		INT	40
C	FOR USE ON THE IBM 360-370 COMPUTER		IBM	41
C			INT	50
	LOGICAL*1 TEST		INT	60
	EQUIVALENCE(TEST,ACHK)		INT	70
	ACHK=WORD		INT	80
	IF(TEST) 10,20,10		INT	90
10	INTGER=.FALSE.		INT	100
	RETURN		INT	110
20	INTGER=.TRUE.		INT	120
	RETURN		INT	130
	END		INT	140

	LOGICAL FUNCTION	INTGER(WORD)	INT	10
C			INT	20
C	THIS SUBROUTINE TESTS THE UPPER NINE BITS OF A WORD FOR ZEROS		INT	30
C	INTGER RETURNS TRUE IF THE WORD IS AN INTEGER OTHERWISE IT RETURNS		INT	40
C	FALSE		INT	50
C	FOR USE ON THE UNIVAC 1100 SERIES COMPUTER		INT	51
C			INT	60
	ITEST=FLD(0,9,WORD)		INT	70
	IF(ITEST)10,20,10		INT	80
10	INTGER=.FALSE.		INT	90
	RETURN		INT	100
20	INTGER=.TRUE.		INT	110
	RETURN		INT	120
	END		INT	130

## RDUSER USAGE

### Control-Card Operation for RDUSER

RDUSER is executed on different computers by a different set of control cards. The following three sets are acceptable for the indicated computer:

(1) CDC 6000 series (Control Data 6000 series computer systems)

```
JOB,...
REQUEST,TAPE9,HY.  number,ROL  (Binary input tape)
REWIND(TAPE9)
REQUEST,TAPE8,HY,X.  number,RIL, initials, identification  (BCD output tape)
    X (external) is optional depending on where the tape is being sent.
RUN(S)
LGO.
DROPFIL(TAPE8)
DROPFIL(TAPE9)
789  (END-OF-RECORD)
    PROGRAM RDUSER(INPUT,OUTPUT,TAPE5=INPUT,
    TAPE6=OUTPUT,TAPE8,TAPE9)
.
. {source deck}
.
789
.
. {card input}
.
6789  (END-OF-FILE)
```

(2) IBM 360-370 series

```
//JOB,...
//A EXEC FORTRAN H
//TIME = number
//SYSIN DD *
.
. {source deck}
.
//B EXEC LINKGO
```

```
//GO.FT05F001 DD *
```

```
.  
· {card input}  
.
```

```
//GO.FT08F001 DD UNIT=7TRACK,VOL=SER=number,  
// LABEL=(,NL),DISP=NEW,DSN=name,  
// DCB=(RECFM=F,LRECL=136,BLKSIZE=136,TRTCH=ET (BCD output tape)  
//GO.FT09F001 DD UNIT=2400-3,VOL=SER=number,  
// LABEL=(,NL),DISP=OLD,DSN=name,  
// DCB=(RECFM=VBS,LRECL=84,BLKSIZE=944) (Binary input tape)  
/*
```

### (3) UNIVAC 1100 series

```
@ RUN,//...  
@ ASG,T      9,T,SAVE05      {Binary input tape}  
@ REWIND     9  
@ ASG,T      8,T,SAVE05      {BCD output tape}  
@ REWIND     8  
@ FOR,IS     RDUSER,RDUSER  
  
·  
· {source deck}  
·  
  
@ MAP,I      relocatable element, absolute element  
@ XQT        absolute element  
  
·  
· {data deck}  
·
```

### Error Messages Output by RDUSER

Messages from RDUSER indicating an error are defined in the following list:

**LABEL (name) WAS NOT FOUND ON TAPE**

User tape label is missing on the input tape (unit 9).

**THE NAME (name) WAS NOT FOUND ON TAPE**

The matrix or table named is missing on the input tape (unit 9).

### Restrictions in RDUSER

The dimensions for DVAL, VAL, IRDIN, and VAL in the two subroutines MATRD and TABRD must be large enough to accommodate a single column of the largest matrix and table being transferred.



The tape must be created on a 7-track tape drive when the receiving installation has a CDC or a UNIVAC computer.

### Card Input for RDUSER

The input for RDUSER is as follows:

The first N cards are read with a free field format (20A4). These cards are used to describe the data blocks being passed between computers. The Nth card must have a blank in column 77 and END in columns 78 to 80 to stop the reading of the comments (see sample input in the next section). The next card is read with the format (2A4) and will have the FORTRAN User Tape Label (left justified). The remaining cards will be read with the format (2A4,2I2). Columns 1 to 8 will have the data block name (left justified). Column 10 will have a code for determining whether a data block is a matrix (1) or a table (2). Column 12 will contain a code determining the option of printing (1) or not printing (0) the elements of the data block.

### Sample Input for RDUSER

A sample input for RDUSER is as follows:

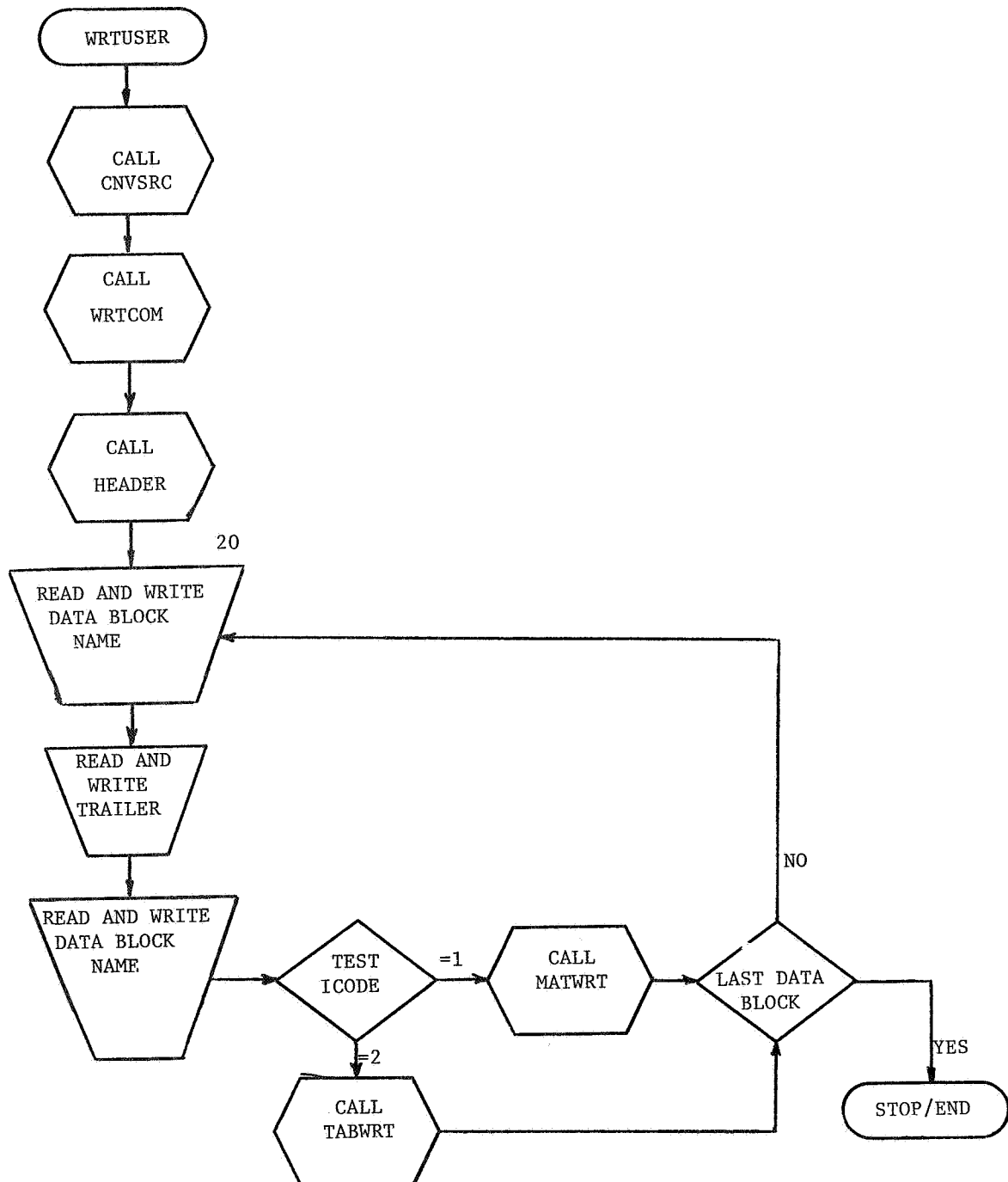
```
THE MATRICES ARE TO TEST SINGLE , DOUBLE PRECISION , COMPLEX AND REAL
      MATRICES , TO TEST THE ORDER OF OUTPUT2 , THE
      ORDER OF HEADERS AND FINALLY
      NULL COLUMNS
END

XXXXXXXXX
ZCOL5      1 1
ZCOL6      1 0
ZCOL8      1 0
ZCOL1      1 0
ZCOL2      1 1
ZCOL3      1 1
B          1 0
X          1 1
EQEXIN     2 1
GPDt       2 0
```

## DESCRIPTION OF WRTUSER AND ITS SUBPROGRAMS

### Program WRTUSER

WRTUSER is a FORTRAN main program that has as its primary function the calling of various subprograms for data manipulation. The flow chart and program listing that follow show how WRTUSER controls program operation.



C		WRT	11
C	THIS PROGRAM CONVERTS A BCD TAPE INTO A BINARY UNFORMATTED TAPE	WRT	12
C	FOR USE AS INPUT INTO NASTRAN MODULE INPUTT2	WRT	13
C		WRT	14
C	DIMENSION NAME(2), ITRAIL(7), IRDIN(100)	WRT	20
C		WRT	21
C	CNVSRC CONVERTS THE SOURCE FROM ONE COMPUTER INTO A SOURCE FORM	WRT	22
C	READABLE BY A DIFFERENT COMPUTER	WRT	23
C		WRT	24
C	CALL CNVSRC	WRT	25
C		WRT	40
C	READ IN COUNTER FOR NUMBER OF DATA BLOCKS FOR THIS LABEL	WRT	50
C	MACH=1 IMPLIES MACHINE FOR OUTPUT TAPE IS CDC	WRT	51
C	MACH=2 IMPLIES MACHINE FOR OUTPUT TAPE IS IBM	WRT	52
C	MACH=3 IMPLIES MACHINE FOR OUTPUT TAPE IS UNIVAC	WRT	53
C		WRT	60
C	READ (5,40) NAM1,NAM2,ITOT,MACH	WRT	70
C		WRT	90
C	CALL SUBROUTINES FOR COMMENT AND HEADER INFORMATION	WRT	100
C		WRT	110
	PRINT 50	WRT	120
	CALL WRTCOM	WRT	130
	PRINT 50	WRT	140
	CALL HEADER	WRT	150
	I=0	WRT	160
20	ICNT=2	WRT	170
	WRITE (8) ICNT	WRT	180
	READ (9,60) NAME,ICODE	WRT	190
C		WRT	200
C	WRITE THE DATA BLOCK NAME	WRT	210
C		WRT	220
	WRITE (8) NAME	WRT	230
	ICNT=-1	WRT	240
	WRITE (8) ICNT	WRT	250
	ICNT=7	WRT	260
	WRITE (8) ICNT	WRT	270
	READ (9,70) ITRAIL	WRT	280
C		WRT	290
C	WRITE THE TRAILER	WRT	300
C		WRT	310
	WRITE (8) ITRAIL	WRT	320
	ICNT=-2	WRT	330
	WRITE (8) ICNT	WRT	340
	READ (9,90) ICNT	WRT	350
	WRITE (8) ICNT	WRT	360
C		WRT	370
C	READ AND WRITE THE DATA BLOCK NAME	WRT	380
C		WRT	390
	READ (9,100) (IRDIN(L),L=1,ICNT)	WRT	400
	WRITE (8) (IRDIN(L),L=1,ICNT)	WRT	401
C		WRT	410
C	CALL A SUBROUTINE TO WRITE THE MATRICES	WRT	420
C		WRT	430
	IF (ICODE.EQ.1) CALL MATWRT(MACH)	WRT	440
C		WRT	450
C	CALL A SUBROUTINE TO WRITE THE TABLES	WRT	460
C		WRT	470
	IF (ICODE.EQ.2) CALL TABWRT	WRT	480
	I=I+1	WRT	490
C		WRT	500
C	CHECK FOR THE END OF THE DATA BLOCKS FOR THAT HEADING	WRT	510
C		WRT	520

PRINT 80, NAME	WRT 530
IF (I.LT.ITOT) GO TO 20	WRT 540
END FILE 8	WRT 560
STOP	WRT 570
C	WRT 580
40 FORMAT (2A4,2I5)	WRT 590
50 FORMAT (1H1)	WRT 600
60 FORMAT (50X,2(A4,6X),I10)	WRT 610
70 FORMAT (10X,7I10)	WRT 620
80 FORMAT (1H0,12H DATA BLOCK ,2A4,25H HAS BEEN WRITTEN ON TAPE)	WRT 630
90 FORMAT (70X,I10)	TAB 631
100 FORMAT (2(A4,6X),(6I10))	TAB 632
END	WRT 640-

## Subprogram CNVSRC

CNVSRC is a FORTRAN subprogram. The primary job of CNVSRC is to convert the source of one computer into a source form readable by a computer of a different type. The CNVSRC subprogram listing follows.

	SUBROUTINE CNVSRC	CONV 10
C		CONV 20
C	THIS PROGRAM CONVERTS THE SOURCE FROM ONE COMPUTER (MACH1) TO THE	CONV 30
C	SOURCE FOR ANOTHER COMPUTER (MACH2)	CONV 40
C		CONV 50
	DIMENSION IDATA(20)	CONV 60
	DATA ISTOP,1T/4HSTOP,4H 1T /	CONV 70
	K=0	CONV 80
C		CONV 90
C	READ IN COMPUTER TYPES	CONV100
C		CONV110
	READ (5,30) MACH1,MACH2	CONV120
	PRINT 40,MACH1,MACH2	CONV130
C		CONV140
C	READ IN DATA	CONV150
C		CONV160
10	READ (3,50) IDATA	CONV170
	IF (IDATA(19).EQ.ISTOP.AND.IDATA(20).EQ.1T) GO TO 20	CONV180
	K=K+1	CONV190
	WRITE (9,50) IDATA	CONV200
	GO TO 10	CONV210
20	PRINT 60,K	CONV220
	REWIND 9	CONV230
	RETURN	CONV240
C		CONV250
30	FORMAT (A4,2X,A4)	CONV260
40	FORMAT( 1H1,34H START CONVERTING SOURCE FROM THE ,A4,17H COMPUTER	CONV270
	1TO THE ,A4,9H COMPUTER /)	CONV280
50	FORMAT (20A4)	CONV290
60	FORMAT (//24H STOP SOURCE CONVERSION ,/1X,15,29H RECORDS HAVE BEEN	CONV300
	1 CONVERTED )	CONV310
	END	CONV320

### Subprogram WRTCOM

WRTCOM is a FORTRAN subprogram that reads the comments generated by RDUSER on unit 9 and prints them out. The WRTCOM subprogram listing follows.

	SUBROUTINE WRTCOM	COM 10
C		COM 11
C	THIS SUBROUTINE READS AND PRINTS COMMENTS	COM 12
C		COM 13
	DIMENSION ICOM(20)	COM 20
	DATA IQUIT/4H END/	COM 30
10	READ (9,20) (ICOM(I),I=1,20)	COM 40
	PRINT 20, (ICOM(I),I=1,20)	COM 50
	IF (ICOM(20).NE.IQUIT) GO TO 10	COM 60
	RETURN	COM 70
C		COM 80
20	FORMAT (20A4)	COM 90
	END	COM 100-



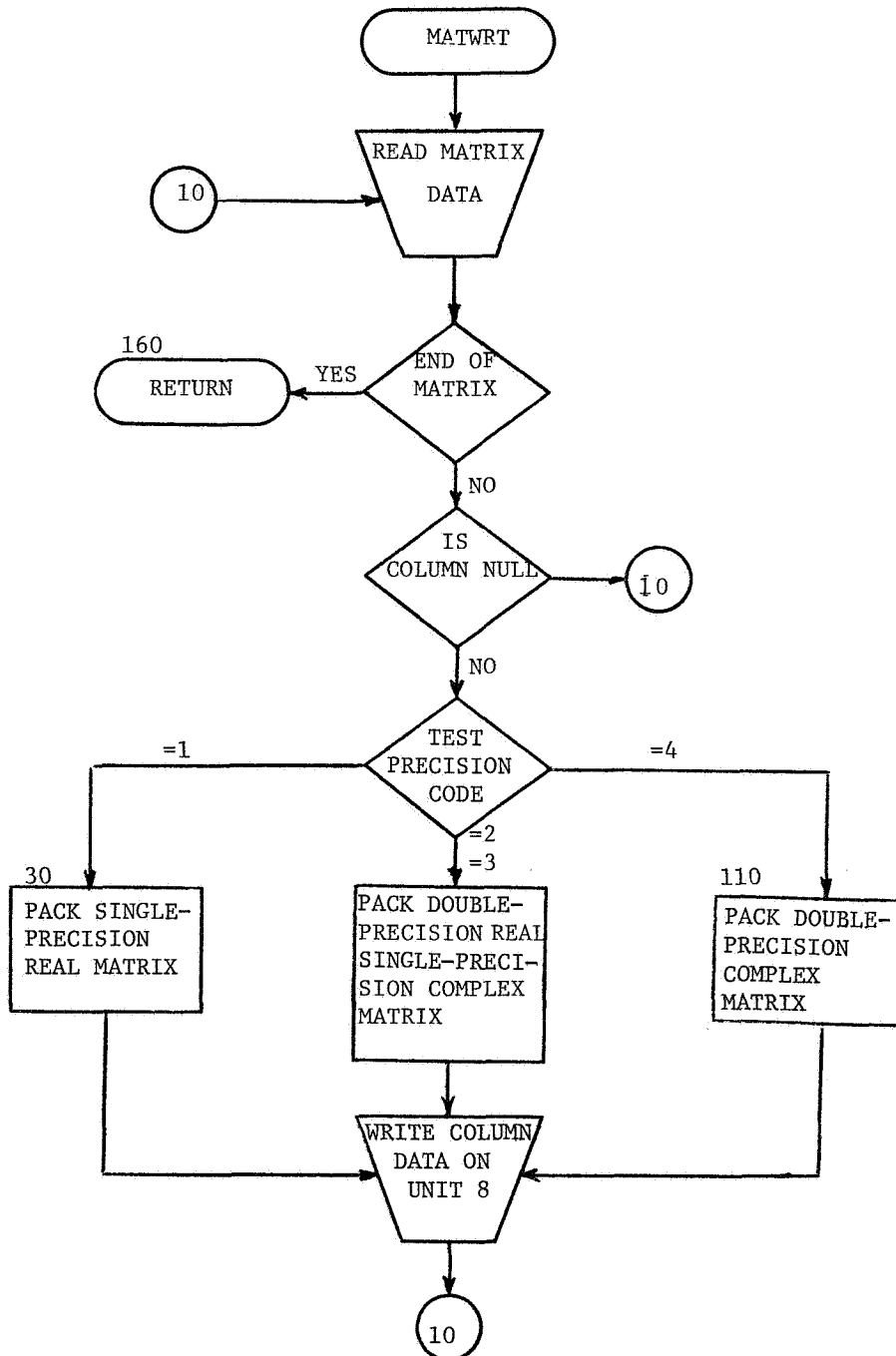
## Subprogram HEADER

HEADER is a FORTRAN subprogram. Its purpose is to write the header information on unit 8 in binary form. The header information includes the data, header, and user tape label. The HEADER subprogram listing follows.

	SUBROUTINE HEADER	HED 10
C		HED 20
C	SUBROUTINE FOR WRITING HEADER INFORMATION	HED 30
C		HED 40
	DIMENSION IDATE(3), HEAD(7), NAME(2)	HED 50
	ICNT=3	HED 60
	WRITE (8) ICNT	HED 70
	READ (9,10) IDATE	HED 80
		HED 90
C	WRITE THE DATE	HED 100
C		HED 110
	WRITE (8) IDATE	HED 120
	ICNT=7	HED 130
	WRITE (8) ICNT	HED 140
	READ (9,20) HEAD	HED 150
		HED 160
C	WRITE THE HEADER	HED 170
C		HED 180
	WRITE (8) HEAD	HED 190
	ICNT=2	HED 200
	WRITE (8) ICNT	HED 210
	READ (9,30) NAME	HED 220
C		HED 230
C	WRITE THE LABEL	HED 240
C		HED 250
	WRITE (8) NAME	HED 260
	ICNT=-1	HED 270
	WRITE (8) ICNT	HED 280
	ICNT=0	HED 290
	WRITE (8) ICNT	HED 300
	RETURN	HED 310
C		HED 320
10	FORMAT (50X,3I10)	HED 330
20	FORMAT (10X,7(A4,6X))	HED 340
30	FORMAT (60X,2(A4,6X))	HED 350
	END	HED 360-

### Subprogram MATWRT

MATWRT is a FORTRAN subprogram. Its parameter is the computer (MACH) on which the tape output on unit 8 from WRTUSER will be used. The primary task for MATWRT is to pack a matrix a column at a time and write it in binary form on unit 8. A MATWRT flow chart and a subprogram listing follow.



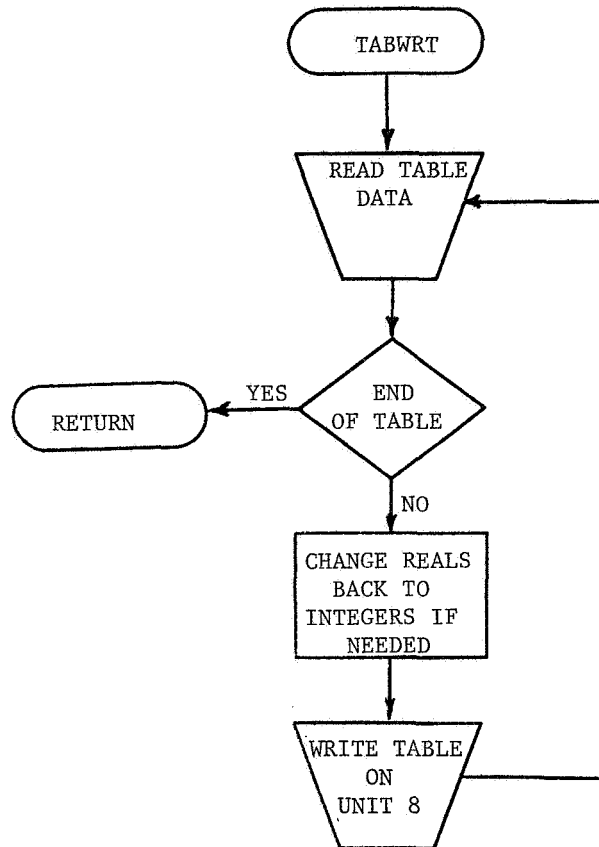
	SUBROUTINE MATWRT(MACH)	MAT 10
C		MAT 20
C	SUBROUTINE FOR WRITING MATRICES	MAT 30
C		MAT 40
C		MAT 41
C	DIMENSION STATEMENT MUST BE ADJUSTED ACCORDING TO SIZE OF MATRIX	MAT 42
C		MAT 43
	DIMENSION IRDIN(5), WRDIN(100), IVAL(200), VAL(200)	MAT 50
	EQUIVALENCE (IVAL(1),VAL(1))	MAT 60
	IF (MACH.EQ.1) ISTORE=131071	MAT 61
	IF (MACH.EQ.2) ISTORE=16777215	MAT 62
	IF (MACH.EQ.3) ISTORE=29843125	MAT 63
	JCNT=-2	MAT 70
10	JCNT=JCNT-1	MAT 80
	WRITE (8) JCNT	MAT 90
	READ (9,170) ICNT	MAT 100
C		MAT 110
C	CHECK FOR END OF DATA BLOCK	MAT 120
C		MAT 130
	IF (ICNT.EQ.0) GO TO 160	MAT 140
	IF (ICNT.NE.1) GO TO 20	MAT 150
	READ (9,170) INUMB	MAT 160
	WRITE (8) ICNT	MAT 170
	WRITE (8) ISTORE	MAT 180
	GO TO 10	MAT 190
20	READ (9,180) (IRDIN(I),I=1,5)	MAT 200
	NZCOL=IRDIN(1)	MAT 220
	IPREC=IRDIN(2)	MAT 230
	READ (9,190) (WRDIN(I),I=1,ICNT)	MAT 240
	KCNT=0	MAT 250
	ISW=0	MAT 260
C		MAT 270
C	IPREC=1 IMPLIES SINGLE PRECISION REAL MATRIX	MAT 280
C	IPREC=2 IMPLIES DOUBLE PRECISION REAL MATRIX	MAT 290
C	IPREC=3 IMPLIES SINGLE PRECISION COMPLEX MATRIX	MAT 300
C	IPREC=4 IMPLIES DOUBLE PRECISION COMPLEX	MAT 310
C		MAT 320
	GO TO (30,70,70,110), IPREC	MAT 330
C		MAT 340
C	WRITE SINGLE PRECISION REAL MATRICES	MAT 350
C		MAT 360
30	DO 60 I=NZCOL,ICNT	MAT 370
C		MAT 380
C	CHECK FOR IMBEDDED ZEROS	MAT 390
C		MAT 400
	IF (WRDIN(I).EQ.0.) GO TO 50	MAT 410
	IF (ISW.EQ.0) GO TO 40	MAT 420
	KCNT=KCNT+1	MAT 430
C		MAT 440
C	STORE POINTER TO NEXT NON-ZERO ELEMENT	MAT 450
C		MAT 460
	IVAL(KCNT)=I	MAT 470
40	KCNT=KCNT+1	MAT 480
	ISW=0	MAT 490
C		MAT 500
C	STORE NON-ZERO ELEMENTS	MAT 510
C		MAT 520
	VAL(KCNT)=WRDIN(I)	MAT 530
	GO TO 60	MAT 540

50	ISW=1	MAT 550
60	CONTINUE	MAT 560
	LCNT=KCNT+6	MAT 570
	WRITE (8) LCNT	MAT 580
C		MAT 590
C	WRITE COLUMN OF MATRIX ON TAPE	MAT 600
C		MAT 610
	WRITE (8) (IRDIN(I),I=1,5),(VAL(I),I=1,KCNT),ISTORE	MAT 620
	GO TO 10	MAT 630
C		MAT 640
C	WRITE DOUBLE PRECISION REAL MATRICES	MAT 650
C	WRITE SINGLE PRECISION COMPLEX MATRICES	MAT 660
C		MAT 670
70	IZCOL=(2*NZCOL)-1	MAT 680
	DO 100 I=IZCOL,ICNT,2	MAT 690
C		MAT 700
C	CHECK FOR IMBEDDED ZEROS	MAT 710
C		MAT 720
	IF (WRDIN(I).EQ.0..AND.WRDIN(I+1).EQ.0.) GO TO 90	MAT 730
	IF (ISW.EQ.0) GO TO 80	MAT 740
	KCNT=KCNT+1	MAT 750
C		MAT 760
C	STORE POINTER TO NEXT NON-ZERO ELEMENT	MAT 770
C		MAT 780
	IVAL(KCNT)=(I+1)/2	MAT 790
80	KCNT=KCNT+1	MAT 800
C		MAT 810
C	STORE NON-ZERO ELEMENTS	MAT 820
C		MAT 830
	VAL(KCNT)=WRDIN(I)	MAT 840
	KCNT=KCNT+1	MAT 850
	VAL(KCNT)=WRDIN(I+1)	MAT 860
	ISW=0	MAT 870
	GO TO 100	MAT 880
90	ISW=1	MAT 890
100	CONTINUE	MAT 900
	LCNT=KCNT+6	MAT 910
	WRITE (8) LCNT	MAT 920
C		MAT 930
C	WRITE COLUMN OF MATRIX ON TAPE	MAT 940
C		MAT 950
	WRITE (8) (IRDIN(I),I=1,5),(VAL(I),I=1,KCNT),ISTORE	MAT 960
	GO TO 10	MAT 970
C		MAT 980
C	WRITE DOUBLE PRECISION COMPLEX MATRICES	MAT 990
C		MAT1000
110	IZCOL=(4*NZCOL)-3	MAT1010
	DO 150 I=IZCOL,ICNT,4	MAT1020
C		MAT1030
C	CHECK FOR IMBEDDED ZEROS	MAT1040
C		MAT1050
	IF (WRDIN(I).EQ.0..AND.WRDIN(I+1).EQ.0..AND.WRDIN(I+2).EQ.0..AND.WRDIN(I+3).EQ.0.) GO TO 140	MAT1060
	IF (ISW.EQ.0) GO TO 120	MAT1070
	KCNT=KCNT+1	MAT1080
C		MAT1090
C	STORE POINTER TO NEXT NON-ZERO ELEMENT	MAT1100
C		MAT1110
C		MAT1120
	IVAL(KCNT)=(I+3)/4	MAT1130
120	ISW=0	MAT1140
	DO 130 J=1,4	MAT1150
	KCNT=KCNT+1	MAT1160

C		MAT1170
C	STORE NON-ZERO ELEMENTS	MAT1180
C		MAT1190
	VAL(KCNT)=WRDIN(I+J-1)	MAT1200
130	CONTINUE	MAT1210
	GO TO 150	MAT1220
140	ISW=1	MAT1230
150	CONTINUE	MAT1240
	LCNT=KCNT+6	MAT1250
	WRITE (8) LCNT	MAT1260
C		MAT1270
C	WRITE COLUMN OF MATRIX ON TAPE	MAT1280
C		MAT1290
	WRITE (8) (IRDIN(I),I=1,5),(VAL(I),I=1,KCNT),ISTORE	MAT1300
	GO TO 10	MAT1310
160	WRITE (8) ICNT	MAT1320
	RETURN	MAT1330
C		MAT1340
170	FORMAT (70X,110)	MAT1350
180	FORMAT (30X,5110)	MAT1360
190	FORMAT (4E20,12)	MAT1370
	END	MAT1380-

### Subprogram TABWRT

TABWRT is a FORTRAN subprogram that reads a table from unit 9, converts real numbers to integers if needed, and writes the table on unit 8 in binary form. A TABWRT flow chart and a subprogram listing follow.





C	SUBROUTINE TABWRT	TAB 10
C		TAB 20
C	SUBROUTINE FOR WRITING TABLES	TAB 30
C		TAB 40
C		TAB 41
C	DIMENSION STATEMENT MUST BE ADJUSTED ACCORDING TO SIZE OF TABLE	TAB 42
C		TAB 43
	DIMENSION IVAL(100),VAL(100),WRDIN(100)	TAB 50
	EQUIVALENCE (IVAL(1),VAL(1))	TAB 51
	JCNT=-3	TAB 60
10	WRITE (8) JCNT	TAB 70
	READ (9,30) ICNT	TAB 80
C		TAB 100
C	CHECK FOR END OF DATA BLOCK	TAB 110
C		TAB 120
	IF (ICNT.EQ.0) GO TO 20	TAB 130
	READ (9,40) (WRDIN(I),I=1,ICNT)	TAB 140
C		TAB 141
C	CHANGE REALS BACK INTO INTEGERS IF NEEDED	TAB 142
C		TAB 143
	KCNT=0	TAB 144
	I=0	TAB 145
11	I=I+1	TAB 146
	IF(I.GT.ICNT) GO TO 13	TAB 147
	IF(WRDIN(I).EQ.-9999.99) GO TO 12	TAB 148
	KCNT=KCNT+1	TAB 149
	VAL(KCNT)=WRDIN(I)	TAB 150
	GO TO 11	TAB 151
12	I=I+1	TAB 152
	KCNT=KCNT+1	TAB 153
	IVAL(KCNT)=IFIX(WRDIN(I))	TAB 154
	GO TO 11	TAB 155
C		TAB 150
C	WRITE THE TABLE	TAB 160
C		TAB 170
13	WRITE (8) KCNT	TAB 179
	WRITE (8) (VAL(I),I=1,KCNT)	TAB 180
	JCNT=JCNT-1	TAB 190
	GO TO 10	TAB 200
20	WRITE (8) ICNT	TAB 210
	RETURN	TAB 211
C		TAB 220
30	FORMAT (70X,I10)	TAB 230
40	FORMAT(4E20,12)	TAB 240
	END	TAB 250-

## WRTUSER USAGE

### Control-Card Operation for WRTUSER

WRTUSER is executed on different computers by a different set of control cards. The following three sets are acceptable for the indicated computer:

(1) CDC 6000 series

```
JOB,...
REQUEST TAPE3,HY,X.  number,ROL  (BCD input tape)
    X (external) is optional depending on where the tape was generated
REWIND(TAPE3)
REQUEST(TAPE8),HY.  number,RIL, initials, identification  (Binary output tape)
REWIND(TAPE8)
RUN(S)
LGO.
DROPFIL(TAPE3)
DROPFIL(TAPE8)
EXIT.
DROPFIL(TAPE3)
DROPFIL(TAPE8)
789  (END-OF-RECORD)
PROGRAM WRTUSER(INPUT,OUTPUT,TAPE5=INPUT,TAPE8,TAPE9,TAPE3)
.
.  {source deck}
.
789  (END-OF-RECORD)
.
.  {card input}
.
6789  (END-OF-FILE)
```

(2) IBM 360-370 series

```
//JOB,...
//A EXEC FORTRANH
//TIME=number
//SYSIN DD *
```

```

. {source deck}
.

//B EXEC LINKGO
//GO.FT05F001 DD *

. {card input}
.

//GO.FT03F001 DD UNIT=7TRACK,VOL=SER=number,LABEL=(,NL),
//  DISP=OLD,DSN=name,DCB=(RECFM=F,LRECL=136,BLKSIZE=136,TRTCH=ET)
    (BCD input tape)
//GO.FT08F001 DD UNIT=2400-3,VOL=SER=number,LABEL=(,NL),
//  DISP=NEW,DSN=name,DCB=(RECFM=VBS,LRECL=84,BLKSIZE=844)
    (Binary output tape)
//GO.FT09F001 DD UNIT=2314,DISP=NEW,SPACE=(TRK,(10,1)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=960)
/*

```

### (3) UNIVAC 1100 series

```

@ RUN, //
@ ASG,T      3,T,SAVE05          (BCD input tape)
@ REWIND     3
@ ASG,T      8,T,SAVE05          (Binary output tape)
@ REWIND     8
@ ASG,T      9,T                (scratch space)
@ FOR,IS     WRTUSER,WRTUSER

. {source deck}
.

@ MAP,I      relocatable element, absolute element
@ XQT        absolute element

. {data deck}
.

```

### Error Messages Output by WRTUSER

There are no error messages in the WRTUSER program.

### Restrictions in WRTUSER

The dimensions for IVAL, VAL, and WRDIN in the two subroutines TABWRT and MATWRT must be large enough to accommodate a single column of the largest matrix and table being transferred.

### Card Input for WRTUSER

The input for WRTUSER is as follows:

The first card, a comment card, is read with the format (A4,2X,A4) where columns 1 to 4 have the type of computer (left justified) on which the tape was created and columns 7 to 10 contain the type of computer (left justified) on which the tape is being converted. The second card is read with the format (2A4,2I5). Columns 1 to 8 have the FORTRAN User Tape Label (left justified). Columns 9 to 13 have the number of data blocks to be converted. Column 18 has a code to show that the tape is being converted for use on a CDC (1), IBM (2), or UNIVAC (3) computer.

### Sample Input for WRTUSER

The following is a sample input for WRTUSER:

```
IBM    CDC
XXXXXXXX 10    1
```

### VERIFICATION OF PROGRAMS

RDUSER and WRTUSER were executed for four<sup>1</sup> of the nine possibilities shown in figure 1 and found to possess the desired qualities lacking in DMI punched cards. Card handling for the input to NASTRAN was cut to the minimum. Square, rectangular, and symmetric matrices containing single-precision real, single-precision complex, double-precision real, and double-precision complex elements were used in the test runs. In each case the answers listed on one computer agreed with the answers listed on dissimilar and similar computers, which indicated that no precision was lost in the transfer.

---

<sup>1</sup>UNIVAC paths were not tested due to errors in the INPUTT2 and OUTPUT2 NASTRAN modules on the UNIVAC computer.

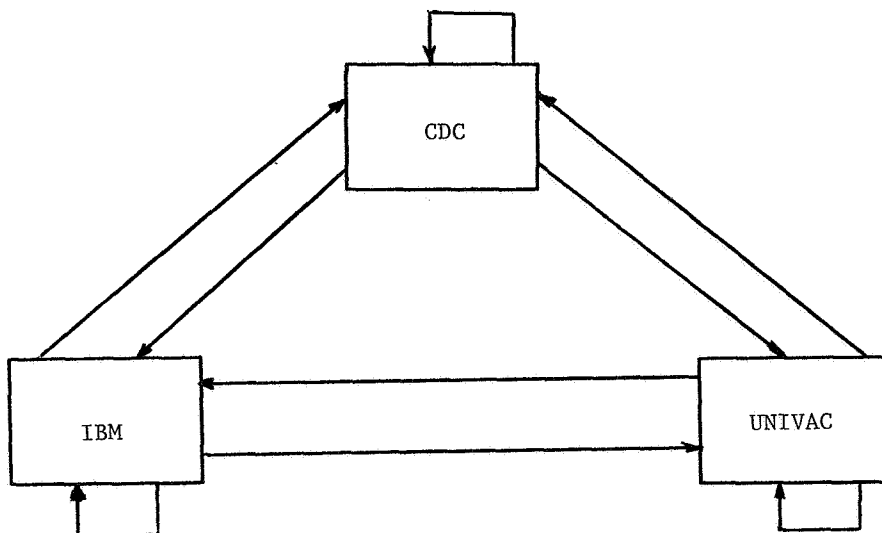


Figure 1.- Paths of data between computers.

### CONCLUDING REMARKS

The transfer of data between dissimilar computers by using the two new NASTRAN utility programs RDUSER and WRTUSER has been done successfully for the IBM and CDC computers. The transfer of data to and from the UNIVAC computer has not been completely tested due to errors in the NASTRAN modules OUTPUT2 and INPUTT2. Square, rectangular, and symmetric matrices containing single-precision real, single-precision complex, double-precision real, and double-precision complex elements were used in the test runs. In each case the answers listed on one computer agreed with the answers listed on a similar or dissimilar computer, which indicated that no precision was lost in the transfer.

Langley Research Center,  
National Aeronautics and Space Administration,  
Hampton, Va., September 24, 1973.

### REFERENCES

1. McCormick, Caleb W., ed.: The NASTRAN User's Manual (Level 15). NASA SP-222(01), 1972.
2. Anon.: The NASTRAN Programmer's Manual. NASA SP-223(01), 1972.

TABLE I.- FORMAT OF HEADER FOR THE NASTRAN USER TAPE

Record	Word	Type (a)	Description
1	1	I	Number of words in next record (3)
2	1 to 3	I	Date
3	1	I	Number of words in next record (7)
4	1 to 7	A	Header
5	1	I	Number of words in next record (2)
6	1 to 2	A	User tape label
7	1	I	End of record (-1)
8	1	I	End of file (0)
9	1	I	Number of words in next record (2)

<sup>a</sup>A = Alphanumeric

I = Integer

TABLE II.- FORMAT OF MATRIX FOR THE NASTRAN USER TAPE

Record	Word	Type (a)	Description
1	1 to 2	A	Matrix label
2	1	I	End of record (-1)
3	1	I	Number of words in next record (7)
4			Trailer
	1	I	Gina name
	2	I	Number of columns
	3	I	Number of rows
	4	I	Form of matrix
	5	I	Type of matrix
	6	I	Number of nonzero terms in the longest record
	7	I	Percent fullness of matrix
5	1	I	End of record (-2)
6	1	I	Number of words in next record (2)
7	1 to 2	A	Matrix label
8	1	I	End of record (-3)
9	1	I	Number of words in next record
10			Column of matrix
	1	I	First nonzero row
	2	I	Precision of matrix
	3 to 5	I	Not used
	6 through no. words -1	R,I	Elements of column. Integer pointers to nonzero elements are imbedded in this record
	No. words	I	End of column
11	1	I	End of record
12 to N			Records 9, 10, and 11 are repeated until a zero (0) is found for the record containing the number of words in the next record
N + 1	1	I	Either a zero (0) for an end of file or a two (2) showing the number of words in the next record

<sup>a</sup>A = Alphanumeric

I = Integer

R = Real



TABLE III. - FORMAT OF TABLE FOR THE NASTRAN USER TAPE

Record	Word	Type (a)	Description
1	1 to 2	A	Table label
2	1	I	End of record (-1)
3	1	I	Number of words in next record (7)
4			Trailer
	1	I	Gino name
	2 to N	I	Miscellaneous information
5	1	I	End of record (-2)
6	1	I	Number of words in next record (2)
7	1 to 2	A	Table label
8	1	I	End of record (-3)
9	1	I	Number of words in next record
10	All	I	Element of first record of table
11	1	I	End of record
12 to N			Records 9, 10, and 11 are repeated until a zero (0) is found for the number of words in the next record
N + 1	1	I	Either a zero (0) for an end of file or a two (2) showing the number of words in the next record

<sup>a</sup>A = Alphanumeric

I = Integer

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS  
PENALTY FOR PRIVATE USE \$300

**SPECIAL FOURTH-CLASS RATE  
BOOK**

POSTAGE AND FEES PAID  
NATIONAL AERONAUTICS AND  
SPACE ADMINISTRATION  
451



POSTMASTER: If Undeliverable (Section 158  
Postal Manual) Do Not Return

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

## NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

**TECHNICAL REPORTS:** Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

**TECHNICAL NOTES:** Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

**TECHNICAL MEMORANDUMS:** Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

**CONTRACTOR REPORTS:** Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

**TECHNICAL TRANSLATIONS:** Information published in a foreign language considered to merit NASA distribution in English.

**SPECIAL PUBLICATIONS:** Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

**TECHNOLOGY UTILIZATION PUBLICATIONS:** Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**  
Washington, D.C. 20546